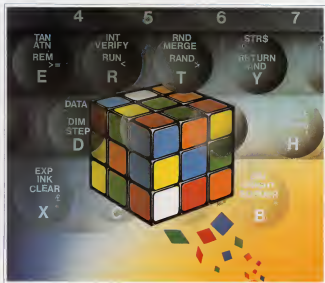


17
150pts.

ARUN

Enciclopedia Práctica del Spectrum



Nueva Lente/Ingelek





PRACTICAS CON MATRICES



SEGURAMENTE lo más complicado del manejo de matrices sea su aplicación práctica. La verdad es que, con los conocimientos adquiridos hasta ahora, podemos emplear de forma eficaz los vectores y tablas en nuestros programas. Sin embargo, para tener una mayor orientación sobre cómo manejar esta potente herramienta de programación, proponemos algunos ejemplos prácticos, tratando de abarcar al menos, las aplicaciones más inmediatas de la misma.

VECTORES

Los vectores o matrices unidimensionales, son la expresión más simple de una estructura de conjunto. Nos encontramos pues, ante una serie ordenada de un determinado número de elementos, calificados por un índice. Decimos que la serie es ordenada porque, con independencia del contenido y la estructura misma de los elementos que la componen, podemos conocer la posición relativa de cualquiera de ellos dentro de la serie, así como cual le precede y cual le sigue.



J\$(8,16)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	J	U	A	N		M	A	N	U	E	L					
2	J	O	S	E		F	E	R	N	A	N	D	O			
3	V	I	C	E	N	T	E									
4	C	A	R	L	O	S										
5	A	N	A													
6	J	O	S	E												
7	M	I	L	A	G	R	O	S								
8	A	N	T	O	N	I	O									

La representación de un vector de cadenas es una tabla.

Supongamos, por ejemplo, un juego en el que intervengan N jugadores. Si almacenáramos sus nombres en una matriz, siempre sabremos a qué jugador le corresponde el siguiente turno de juego, sólo con sumar 1 al índice del vector, y controlando el hecho de hacer este igual a 1 cuando, por efecto de la operación de suma, su valor se haga igual a $N+1$, es decir, exceda al último jugador.

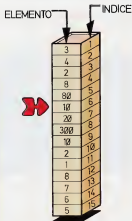
```

10 REM - TURNO DE JUGADORES
20 INPUT "Numero de jugadores:";N
30 DIM J$(N,16)
40 FOR I=1 TO N
50 INPUT "Jugador "I+STR$ N+": "LINE
J$(I)
60 NEXT I
70 PRINT "TURNOS": "JUGADORES"
80 FOR I=0 TO 31: PRINT " ": NEXT I
90 PRINT
100 FOR I=1 TO N: PRINT I;J$(I): NEXT I

```

Las matrices operan al BASIC en sistemas altamente eficientes para el almacenamiento y clasificación de datos.

Una matriz es una serie ordenada, en la cual, independientemente de su contenido, podemos conocer analizando el índice, la posición relativa que ocupan sus elementos.



i!

La hipótesis cuarta dice que, al ser llevada por la acción física, puede ser directamente imaginada o visualmente mediante una estructura tridimensional. Su representación que muestra es evidentemente bastante compleja, aunque generalmente se le asigna el sentido de tiempo a la cuarta dimensión de forma que estas matrices pueden presentarse mediante estructuras espaciales (tridimensionales) evaluadas en un determinado instante (corta dimensión).

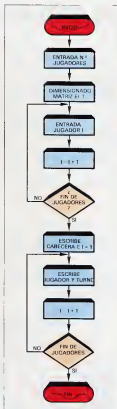
*

El vector es una serie ordenada porque con independencia del contenido de los elementos que lo componen, sabemos la posición relativa de uno dentro de la serie de ellos. Así mismo podemos conocer cuál es el que le precede y el que le sigue.

MATRICES NUMERICAS DE UNA DIMENSION

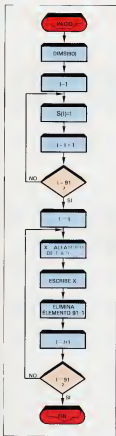
Las matrices pueden definirse como multidimensionales. En cualquier caso, la estructura más simple de matriz numérica es el vector, es decir, la matriz unidimensional.

En base a los conocimientos adquiridos en el capítulo anterior, proponemos ahora el desarrollo de un programa que, a partir de una serie de un número variable de elementos, gestione la extracción al azar de la totalidad de ellos, sin repetición. En el programa de ejemplo, hemos definido la variable N como 80, para simular las extracciones de bolas en el juego del bingo.



Organigrama de la rutina para de jugadores

Organigrama de la rutina de extracción aleatoria.



```

10 REM EXTRACCION ALEATORIA.
J.M. LOPEZ MARTINEZ
20 LET N=90: DIM S(N)
30 FOR I=1 TO N: LET S(I)=I: NEXT I
40 FOR I=1 TO N
50 LET X=INT(RND*(N-I+1))
60 PRINT I: " "STR$(S(X))\LEN STR$(S(X)) TO "
70 FOR J=X TO N-I: LET S(J)=S(J+1)
NEXT J
80 NEXT I
  
```

En la línea 20, se define el número de elementos de la serie de aleatorios a producir, en nuestro caso 90. A continuación, dimensionamos un vector numérico $S(\)$, por medio de la sentencia DIM, con un número N de elementos. Esta lista contendrá la serie ordenada desde el 1 hasta el valor que asignemos de comienzo a la variable N . En la instrucción 30, se efectúa la carga de la matriz $S(\)$, siendo el contenido de cada elemento idéntico al del índice que la carga, es decir, coincide el valor del elemento en cuestión con su posición relativa dentro del vector. Así por ejemplo, el elemento uno contiene el valor uno, el dos el valor dos, etc.

En las líneas 10 a 80 se encuentra el bucle central del programa, establecido para los valores de I comprendidos entre 1 y N , y en la línea 50, se efectúa la extracción aleatoria de un elemento de la serie, a través de la función RND, calculado con la fórmula siguiente: $LET X=INT(RND*(N-I+1))$.

Antes de comentar en detalle el por qué del empleo de esta fórmula, debemos hablar de las posibilidades que se nos ofrecen a la hora de extraer, de una serie de elementos, algunos de ellos, sin que éstos puedan repetirse.

La primera solución es la propuesta en el programa de ejemplo del capítulo anterior, donde a cada nuevo elemento seleccionado se le hace una marca en una posición determinada para, posteriormente, ejecutar una generación aleatoria controlada. Este control consiste en repetir una y otra vez la generación aleatoria, hasta que arroje como resultado un elemento de la matriz sin marcar, es decir, hasta el momento.

Lógicamente, a ninguno se nos escapa que este método es únicamente eficaz cuando, sobre una serie grande, se han de extraer unos pocos elementos, como es el caso de las diez preguntas sobre el total de los países europeos, esto es debido a que cada vez se genera un número aleatorio, comprendido en el intervalo que va desde uno al total de países europeos. Si en nuestro ejemplo actual empleáramos el mismo procedimiento, según se van extrayendo los números, se hace más y más difícil, obtener uno que no haya salido ya.

Con el empleo de la fórmula de la línea 50, el margen se reduce progresivamente desde N ha-



Una matriz de dos dimensiones es un conjunto de conjuntos, es decir una estructura formada por varios conjuntos de un nivel superior representados por el primer índice, y varios de nivel inferior representados por el segundo.



Al hablar de matrices de las dimensiones dos o más, nos referimos a un conjunto que abarca a un subconjunto, y esto a su vez a otro subconjunto de él, de forma anidada y perfectamente estructurada.



ta uno, por cada nueva pasada del bucle principal de programa FOR NEXT, obteniéndose mejores resultados. Sin embargo, supongamos que el primer número seleccionado no sea el último de la serie (al menos en nuestro caso), cosa por otra parte harto probable, si reducimos en uno el margen de operación de RND (de 1 a 89), descartaría la elección del 90, con lo cual este sistema es auténticamente castrófico, puesto que sólo funcionará cuando el número elegido en cada ocasión sea el último del intervalo (80, 88, 88, 87, etc...). La explicación de por qué elegirá este método, aparentemente absurdo, se encuentra en la línea 70, estudiada un poco más adelante, la cual sirve de complemento a la línea de extracción aleatoria, y hace que el nuevo sistema sea completamente válido.

En la línea 80 se imprime el elemento de la serie obtenido en la línea anterior, justificado por la derecha en cuatro posiciones, cosa que hacemos para que los números queden columnados y aparecen 8 en cada línea. El algoritmo empleado para conseguirlo es: PRINT (" " + STR\$ S(X)(LEN STR\$ S(X) TO 1).

Generalizando, para todas las ocasiones en que nos interese obtener columnas de números ajustados a la derecha, podemos decir que el sistema consista en montar una cadena, suma de tantos espacios como deseenos sea el ancho de la columna menos 1, más la STR\$ de la variable a imprimir, para escribir de toda ella la parte com-

prendida desde la LEN de la STR\$ al final de la cadena. ¿Complicado? Menos de lo que parece, sólo tenemos que practicar un poco con el algoritmo, para llegar a comprender y manejarlo a la perfección.



La estructura matricial puede ser entendida matemáticamente como un conjunto de subconjuntos.

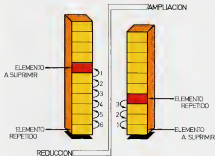
Como ya hemos dicho antes, en la línea 70 incluimos la rutina que asegura la validez del sistema de extracción aleatoria. Esta rutina, en esencia, elimina el elemento elegido de la serie, desplazando todos los demás que se encuentran después de él a la posición relativa anterior. De esta forma, la serie es cada vez más corta, manteniéndose siempre compuesta únicamente por elementos que todavía no han salido.

Si, por ejemplo, el primer número extraído es el tres, los elementos del 4 al 90 (fin del primer intervalo), pasarán a ocupar la posición anterior al cuatro la tres, el cinco la cuatro, hasta el ochenta y nueve. Al mismo tiempo, esta línea nos ha servido para ilustrar la técnica de desplazamiento en elementos de una tabla, utilizada muy frecuentemente, tengamos en cuenta en este sentido, que el algoritmo empleado debe ser diferente según el intervalo se reduzca (programa de ejemplo) o se amplíe.

En este último caso, sólo varía el orden en que se sitúan los elementos, que en vez de ascendente debe ser descendente. Supongamos que queremos desplazar las líneas de una matriz S(), desde la línea 3 a la 89, para que ocupen las posiciones de la 4 a la 90, la instrucción adecuada sería: FOR I= 90 TO 4 STEP -1: LET S(I)=S(I-1). NEXT I.

Si no hubiera sido por querer otorgar un valor más pedagógico al ejemplo, podríamos haber utilizado un sistema bastante más sencillo para dar validez a la generación aleatoria: simplemente, sustituir el elemento extraído por el último del intervalo. Así, siguiendo el ejemplo anterior, si el número extraído hubiera sido el tres, y el último

Según el desplazamiento de elementos de una matriz puede ser reducirlo o ampliarlo, el STEP del FOR empleado en el algoritmo, deberá ser negativo o positivo, respectivamente.





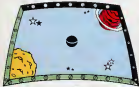
del intervalo el noventa, el problema se hubiera limitado a efectuar `LET S(3)=S(90)`, y generalizando, la línea 70 del programa, pasaría a ser 70 `LET S(X)=S(N-I+1)`.

Por último, en la línea 80, se cierra el bucle principal del programa con la instrucción `NEXT I`. De esta forma, el bucle de la línea 70 queda anidado dentro del bucle exterior.

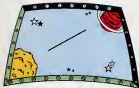
Como podemos ver, el hecho de haber definido el número de elementos de la serie mediante una variable (N), nos permite alterarla fácilmente. Reajustando así el programa, puesto que funciona con 90 elementos (ejemplo), como sería el caso de la extracción de bolas de bingo como con 40, para la extracción de cartas de la baraja española, o cualquier otra aplicación similar.

MATRICES NUMERICAS DE DOS DIMENSIONES

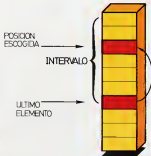
Para tener una idea clara de la estructura de una matriz de dos dimensiones debemos pensar en



Las variables o de matrices unidimensionales de un solo elemento, pueden representarse geométricamente por un punto.



La representación geométrica de líneas (vectores) es una línea.



El sistema más fácil para dar salida al algoritmo de extracción aleatoria, habría sido hacer una copia en la posición elegida del último elemento del intervalo.

que nos movemos dentro de un conjunto de conjuntos, es decir, tenemos varios conjuntos de un nivel superior, representados por el primer índice, y varios de nivel inferior, representados por el segundo.

Dado que para acceder a un elemento determinado debemos especificar sus dos índices, vemos claramente una analogía con la representación cartesiana en el plano, a través de la abscisa y la ordenada, debido a ello, este tipo de variables suscritas se conocen como TABLAS, puesto que gráficamente pueden ser representadas como una tabla formada por filas y columnas, o si lo preferimos, como una serie de vectores situados uno al lado de otro.

A modo de ejemplo, proponemos un programa por el que se obtiene un informe sobre las calificaciones y notas medias de los 9 alumnos de una clase.

```

10 REM ESTADISTICA DE CALIFICACIONES
J.M LOPEZ MARTINEZ
20 DIM C(9,9)
30 PRINT TAB 7: "- CARGA DE DATOS -"
40 FOR L=0 TO 3: PRINT " " ; NEXT L
50 PRINT " "
60 FOR J=1 TO 9: FOR K=1 TO 9
70 INPUT "Alumno "+STR$ J+" NOTA "+STR$
K+" : ";C(J,K) IF C(J,K)<0 OR C(J,K)>10
THEN GO TO 70
80 PRINT " " ; STR$ C(J,K) ; LEN STR$
C(J,K) TO 1:
90 NEXT K: PRINT " " ; NEXT J
100 CLS : PRINT "NUM" ;

```

!!

Las matrices o matrices unidimensionales son la expresión más simple de una estructura de conjunto. Consisten en una serie ordenada de un número determinado de elementos etiquetados por un índice.

*

La única limitación de memoria en las matrices la impone la propia memoria por tanto podemos crear estructuras de datos de extraordinaria complejidad.



En general, cualquier estructura matricial, se puede entender como un árbol invertido, en el que cada índice genera las ramas correspondientes a otros índices



Las estructuras matriciales de dos dimensiones, son representadas usualmente como tablas, en las cuales uno de los índices son las filas y el otro las columnas, por tanto, forman geométricamente un plano

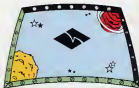


También es usual representar las estructuras matriciales como conjuntos de subconjuntos

```

110 FOR K=1 TO B: PRINT (" N°+STR$ K)\EL
    EN STR$ K TO 3); NEXT K
120 PRINT " N,M"
130 FOR L=8 TO 31: PRINT "="; NEXT L
140 FOR J=1 TO Y
150 LET S=M: PRINT (" "+STR$ J)\LEN ST
    RS J TO 3);
160 FOR K=1 TO B
170 LET R=M+(K-J,K); PRINT (" "+STR$ C(
    J,K)\LEN STR$ C(J,K) TO 3);
180 NEXT K
190 LET X=STR$ INT ((S/R+.85)*18): LET
    X=(
    " "+X)\LEN X TO 3); PRINT X\ TO
    3); " "X\LEN X"
200 NEXT J
    
```

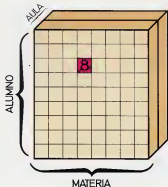
En la línea 20 se define, por medio de una instrucción DIM, la matriz numérica de dos dimensiones C(B,B). Al corresponder un índice por dimensión, nos referenciamos en adelante al primero de ellos (el número de alumno) como J, y al segundo (el número de nota de cada alumno) como K. La matriz tiene, pues, una estructura de nueve filas por ocho columnas, accediéndose a cada elemento de la misma, indicando los correspondientes valores de J y K (alumno y nota). En las líneas 30 a 50 se gestiona la cabecera de la carga de datos. Primero se imprime un título centrado y, seguidamente, una línea de guiones tras la cual se produce una de separación. En las líneas 60 a 80 se gestiona la entrada de datos propiamente dicho. La 60 abre los bucles FOR NEXT anidados de las variables J y K, los cuales realizan la carga ordenada de los datos en la tabla, gracias a la siguiente instrucción, que contiene el INPUT



La representación gráfica empleada para una matriz bidimensional, es un plano

La instrucción 70 se divide en dos partes bien diferentes. En la primera de ellas, se aceptan los datos con un mensaje de petición, construido a base de las palabras descriptivas de los índices y los respectivos valores de éstos mismos, conociéndose de este modo, en todo momento, el dato exacto a solicitar. En la segunda parte, se efectúa la depuración mínima necesaria para obtener una cierta garantía sobre la validez de los datos introducidos. En este caso, como el INPUT es de tipo numérico, no es necesario verificar que los datos introducidos lo sean. De esta forma, con

El solo es el tercer índice utilizando en el programa de ejemplo para determinar un elemento de la matriz.



ALUMNO 3	AULA 1
<u>CALIFICACIONES</u>	
1 MATEMATICAS	9
2 FISICA	2
3 QUIMICA	4
4 BIOLOGIA	8
5 LITERATURA	5
6 LENGUA	7
7 INGLES	9
8 FILOSOFIA	3

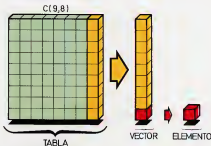
contramos nuestros esfuerzos en controlar que datos se encuentren dentro de los márgenes razonables (0 a 10)

En la línea 80 se van imprimiendo los datos introducidos justificada a la derecha, dentro de columnas de 3 posiciones cada una, completando así una línea por cada 8 calificaciones introducidas.

La línea 90 cierra los bucles anidados de notas y alumnos, controlando un espacio doble entre líneas por medio de la instrucción PRINT "", con el fin de en esta línea la carga de datos en la misma.

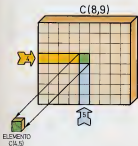
Las líneas de la 100 a la 200 gestionan la impresión de la estadística de resultados obtenidos. Dentro de ellas, las líneas 100 a 130 imprimen una cabecera para los resultados. Esta cabecera se construye, partiendo del borrado de la pantalla y la impresión de un título, para la columna destinada a albergar los números de los alumnos. A continuación, se imprimen una serie de ocho cabeceras formadas por la inicial N seguida del número de la columna. Por último, retornamos a la misma rutina de subrayado con el símbolo (-), ya empleado en ocasiones anteriores.

En las líneas 140 a 200 se desarrolla verdaderamente la actividad de impresión de calificaciones y cálculo e impresión de la nota media. La 140 comienza con el bucle exterior de alumnos, seguidamente, la instrucción 150 imprime el comienzo de la línea de detalle de calificaciones de cada alumno, consistente en su número centrado frente



La matriz anterior puede entenderse como conjunto de vectores verticales (tabla). Tal es el caso de la matriz C(9,8) en el programa de ejemplo.

Para designar cualquier elemento de una tabla, son precisos dos índices:



tn a la cabecera, haciendo cero la variable S, que contiene la suma de las notas de cada alumno para el cálculo de la media.

En las líneas 160 a 180 codificamos el bucle de impresión de calificaciones columnadas por la derecha, con acumulación en la variable S de cada nota.

Por último, en la instrucción 190 resolvemos la impresión de la nota media, después de efectuar su cálculo. La operación es simple, bastando con dividir S por ocho. Sin embargo, nosotros la hemos complicado un poco, al asegurarnos de que las notas medias se impriman redondeadas y con un decimal. Para poder hacerlo, utilizaremos el algoritmo: $INT((S/8+.05)*10)$.

Este algoritmo puede utilizarse de forma general para imprimir los valores numéricos redondeados a un número fijo de decimales. Para ello, debemos tener en cuenta que a la variable en cuestión es preciso sumarle un 5 en la posición decimal siguiente por la derecha a la que pretendemos redondear el resultado. En nuestro ejemplo, sumamos .05 porque queremos redondear a un decimal. Si quisiéramos redondear a enteros escribiríamos .5 y para redondear el resultado a dos decimales .005.

Una vez hecho esto, basta con multiplicar el resultado obtenido por 10, 100, ... para obtener uno, dos ... decimales, y por último, calcular la parte entera de lo obtenido hasta el momento. Con esto concluye la fase de cálculo, pero aún falta imprimir los resultados, es decir, interpretar el contenido de la variable aforementioned en la cual hemos colocado el resultado de las operaciones, en nuestro caso X8, cuyo valor es la STR\$ del INT del valor numérico obtenido.

i!

Para acceder a un elemento determinado debemos especificar los dos índices que definen una clara analogía con la representación cartográfica en el plano a través de la abscisa y la ordenada.

Los matrices tridimensionales precisan de tres índices para acceder a cada elemento, lo que implica cierta relación con la representación espacial para, en los dimensiones.

Ya citándonos a nuestro ejemplo, en la segunda mitad de la línea 190, efectuamos la impresión de la media con dos instrucciones. En la primera de ellas, ajustamos la variable X9 con tres espacios por su izquierda, antes de comenzar su interpretación, hecho esto, imprimimos la parte entera (tres primeras posiciones), un punto, y el decimal (la cuarta posición): LET X9=" " +X9\$LEN X9 TO 1. PRINT X9\$ TO 3;"."X9(4)"

MÁTRICES NUMERICAS DE TRES DIMENSIONES

Las matrices tridimensionales precisan de tres índices para calificar a cada elemento. Si hemos dicho que las bidimensionales, tenían cierta similitud con la representación cartesiaca en el plano, podemos decir ahora que éstas tienen con la representación espacial pura, en tres dimensiones. Nos movemos, en este caso, en un conjunto que abarca a un subconjunto, y éste a su vez a otro subconjunto de él, de forma ágil y perfectamente estructurada.

Volviendo al ejemplo anterior de las calificaciones, podemos proponer ahora una aplicación al problema planteado anteriormente, donde el análisis se extiende a las dos aulas que componen el curso. Derivamos de esta forma una matriz tridimensional C(2,8,8), e índices I, J y K. Por lo demás, el desarrollo del programa es bastante similar al anterior, con la inclusión de una rutina de borrado parcial de la pantalla, para admitir separadamente las calificaciones de las dos aulas, tal como podemos observar en el código pendiente listado.

```

10 REM ESTADISTICA DE CALIFICACIONES -
  J.M. LOPEZ MARTINEZ
20 DIM C(2,9,8) : LET S8="" : FOR L=0 TO 31: LET S8=S8+" " : NEXT L
30 PRINT TAB 7;"- CARGA DE DATOS -"
40 FOR L=0 TO 31: PRINT " " : NEXT L
50 PRINT
60 FOR I=1 TO 2: PRINT AT 2,0: FOR L=0 TO 16: PRINT S8: NEXT L: PRINT AT 2,0: FOR J=1 TO 9: FOR K=1 TO 8
70 INPUT "Curso "+STR$ I;" alumno "+STR$ J;" NOTA "+STR$ K+" : C(I,J,K): IF C(I,J,K)<0 OR C(I,J,K)>10 THEN GO TO 70
80 PRINT (" " +STR$ C(I,J,K))$LEN STR$ C(I,J,K) TO 1)
90 NEXT K: PRINT " " : NEXT J: PAUSE 0: NEXT I
  
```

```

100 CLS : PRINT "NUM":
110 FOR K=1 TO 8: PRINT (" " +STR$ K)$LEN STR$ K TO 1) : NEXT K
120 PRINT " N,M"
130 FOR L=0 TO 31: PRINT " " : NEXT L
140 FOR I=1 TO 2: PRINT AT 2,0: FOR L=0 TO 16: PRINT S8: NEXT L: PRINT AT 2,0: FOR J=1 TO 9
150 LET S8="" : PRINT (" " +STR$ J)$LEN STR$ J TO 1)
160 FOR K=1 TO 8
170 LET S8=S8+C(I,J,K) : PRINT (" " +STR$ C(I,J,K))$LEN STR$ C(I,J,K) TO 1)
180 NEXT K
190 LET X8=STR$ INT ((S8+.05)*10) : LET X9=" " +X8$LEN X8 TO 1 : PRINT X8: TO 3) : " "X9(4)"
200 NEXT J: PAUSE 0: NEXT I
  
```

Para finalizar añadiremos que las matrices no presentan una restricción a tres dimensiones, sino que también podemos construir estructuras tetradimensionales, etc. Así por ejemplo, podríamos conservar en una misma matriz las notas obtenidas por todos los alumnos en todas las asignaturas para los diferentes cursos, y además, durante todo el curso; en este caso, la cuarta dimensión asociada sería el tiempo (los meses del curso).



Las representaciones tetradimensionales son extraordinariamente más complejas que las vistas hasta ahora; un caso habitual es considerar el tiempo como la cuarta dimensión.



Las matrices tridimensionales son representadas geométricamente con volúmenes.



Las estructuras matriciales unidimensionales (vectores o listas), se pueden representar como una línea, o como un punto en el caso particular de un vector en un solo elemento.



Las estructuras matriciales tridimensionales se pueden representar gráficamente como un volumen en el cual uno de los índices corresponde a la altura, otro a la anchura y el restante a la profundidad. Así pues, representan una figura en el espacio y no en el plano.

LAS IMPRESORAS DEL SPECTRUM



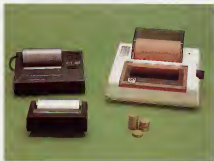
ON el paso del tiempo, entre los usuarios del Spectrum circula una idea generalizada: «A mi ordenador le falta algo». Debido a esto, tanto

Sinclair, fabricante de nuestro micro, como otras firmas dedicadas a la producción de hardware lanzan constantemente al mercado nuevos periféricos, con los que satisfacen la creciente demanda de los usuarios, eso sí, a costa de aumentar la confusión por parte del comprador, dado el elevado número de complementos disponibles. En las páginas de esta sección, tratamos de orientar a todos aquellos que en determinado momento deben tomar una decisión sobre este o aquel equipo accesorio a nuestro Spectrum, así como a los que puedan ya poseerlo, con el fin de esclarecer sus posibilidades de manejo, yendo más allá de las escasas explicaciones que adjuntan los manuales de instrucciones que, por si además fuera poco, en muchos casos se encuentran en inglés.

Cuando tratamos el tema de los joysticks, vimos como, inevitablemente, fuera cual fuera el modelo elegido, debíamos disponer de un interfaz. Lo mismo ocurría con los discos, incluso en el capítulo dedicado como introducción a las impresoras, hacíamos hincapié en los interfaces más utilizados por estas.

Sin duda es la impresora el periférico más ampliamente difundido, entre los ordenadores a los que se les pretende encomendar una misión más allá del genéricoo extraterrestre. Debido a ello, tres firmas comerciales han decidido lanzar al mercado unas impresoras que, sin necesidad de interfaz especial, puedan ajustarse a las necesidades básicas del usuario del Spectrum. Todas ellas con una característica común: se bajó coste. No quiere decir esto que no dispongan de interfaz. Este va acoplado dentro del propio equipo, y además, está diseñado para funcionar exclusivamente con nuestro Spectrum, gracias a lo cual, el precio del periférico disminuye sensiblemente. Las tres impresoras mencionadas, disponen del juego de caracteres completo de nuestro ordenador, incluyendo mayúsculas, minúsculas, caracteres gráficos, en video inverso, etc. Por tanto, admiten la posibilidad de conseguir caracteres subrayados o acorados, de la misma manera que lo haríamos para presentarlos en la pantalla del televisor o monitor.

Para su control, se utilizan los comandos LIST y LPRINT, equivalentes a LIST y PRINT, aunque



Las tres impresoras que tratamos en este capítulo, tienen en común su bajo precio.

referidos a la impresora y no a la pantalla. En este aspecto, es conveniente hacer una pequeña puntualización. La salida de información en el Spectrum, se efectúa a través de dos pares de corrientes, así, las corrientes 0 y 1 están asociadas a la parte inferior de la pantalla, la 2 a su parte superior, y la 3 a la impresora. Las restantes, no se encuentran asignadas a ningún canal (periférico) específico.

Según el periférico que deseemos emplear, el Spectrum se encarga de seleccionar la corriente adecuada. De esta manera, PRINT habitualmente se remite a la zona superior de la pantalla, po-

Los periféricos (CANALES) se envían al ordenador a través de CORRIENTES.





La lentitud y el polvo son los grandes enemigos de nuestra perforadora

demo modificar este estado de cosas, simplemente indicando la corriente concreta por la cual deseamos enviar la información, mediante PRINT # <número de corriente>. Por tanto, la derivación de datos a la impresora, puede conseguirse con PRINT #3, o lo que es lo mismo: LPRINT es una forma abreviada de escribir PRINT #3.

Si queremos compatibilizar un programa, de forma que escriba sus resultados en la impresora en vez de en la pantalla, bastará con ejecutar previamente la orden OPEN #2, "p", gracias a la cual, el flujo que habitualmente se envía a la pantalla (corriente 2 asociada al canal "a"), se bifurca a la impresora, tal como indica el parámetro "p" ("p" como abreviatura de printer) a continuación de OPEN. Volvamos ahora al tema fundamental del capítulo.

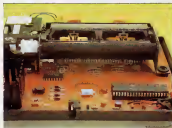
Todos los impresoras que estamos tratando implementan el comando COPY, el cual efectúa

una copia en papel del contenido de la pantalla, sin tener en cuenta, lógicamente, el color de la misma, sino sólo el estado del archivo de imagen, no incluyendo las dos últimas filas, reservadas por lo general al Sistema. Por otra parte, al ser perforadoras especialmente diseñadas para el Spectrum, todas ellas siguen el formato de impresión en treinta y dos columnas por fila.

UNA CHISPA IMPRESIONANTE

Bueno, lo cierto es que no se trata de una enorme descarga eléctrica, que puede hacer peligrar nuestra integridad física en el momento de perfil en funcionamiento la impresora. Simplemente, es el sistema empleado por la ZX PRINTER (impresora ZX) en la composición de los caracteres. Se trata de una impresora electrónica, es decir, cada punto de los que conforman un carácter, se genera al incidir sobre el papel una diminuta chispa eléctrica, producida entre una placa fija, cargada, y una pequeña aguja que se desliza a lo ancho del papel.

El papel utilizado es especial, recubierto en uno de sus caras por una fina capa de compuestos metálicos, de forma que al recibir la descarga, «revela» un punto, cambiando a una tonalidad más oscura. Su anchura máxima es de 100 micras. En el caso de la ZX PRINTER, el elemento impresor no es en realidad una sola aguja, sino dos, que realizan idéntico cometido. Estas se sitúan en oposición una de la otra, sobre una pequeña



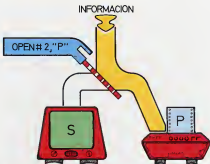
Debido a su sistema de impresión, la Alphabet 32 pertenece al tipo denominado «jet de tinta».



Como se puede observar en la foto, la impresora Siskatek GP-50 es del tipo «matricial».

como dentada de material plástico, apoyada sobre dos engranajes, uno de los cuales va unido al motor que suministra a la correa una velocidad de giro constante, desplazando de esta manera las agujas horizontalmente sobre el papel. Al mismo tiempo, cada vez que una de las agujas completa una pasada, un eje, igualmente engranado al motor, se encarga de desplazar hacia arriba el papel, con el fin de posicionar la siguiente línea de impresión. El sistema resulta, sin duda, ingenioso, y daña la simplicidad del mismo, los resultados, mejores de lo que cabría esperar, siempre y cuando nuestro objetivo no sea otro que el de obtener cortos listados, o sacar algún que otro comentario por impresora.

Moviéndonos en el tiempo de los pequeños detalles, diremos que el nivel del ruido es bajo, y en ningún caso se hace molesto. La calidad de la impresión la hace cuando menos legible, aunque se observan algunas deficiencias a la hora de trazar líneas verticales. Hoy en día, esta impresora que fue una gran ayuda, en su momento, para los piceros del Spectrum, está prácticamente en desuso, dejando paso a otros equipos que la superan en velocidad y calidad.



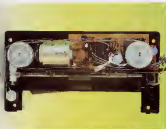
El flujo de información que habitualmente discurre por PRINT hacia la pantalla (canal "5"), puede ser fácilmente derivado a la impresora con OPEN #2, "P" (canal "7")

CARACTERES CALIENTES

Otra posible solución económica a la hora de adquirir una impresora, la proporciona la ALPHA-COM 32. De nuevo, se trata de un aparato compacto, de fácil manejo, totalmente compatible con

nuestro Spectrum. Todo el equipo va encerrado dentro de una carcasa de plástico inyectado, especialmente resistente a los golpes. De él parte el conector adaptable a la tarjeta de expansión de nuestro micro. Para su funcionamiento, es necesaria una fuente de alimentación externa, suministrada junto con la impresora.

Posee un pulsador OFF, cuya misión es desactivar el flujo de información hacia la impresora, aunque esto no, quiere decir que quede desconectada de la alimentación. El botón ON/



El mecanismo de impresión de la ZX Printer, se basa en dos agujas montadas en posición opuesta sobre una póliza



Cada una de las impresoras citadas emplea un tipo de papel diferente: normal (Sordalux GP-58), térmico (AlphaCom 32) y electrónico (ZX Printer)

!

El papel utilizado por las impresoras electrostáticas y térmicas es muy sensible al frotto provocándose por esta causa dificultades en la impresión.

*

Es importante utilizar papel de calidad adecuada para conseguir una buena impresión.

*

En las impresoras Seiko GP 505 y Alphacom 32 debido a la distancia separada entre puntos no resistentes y ventosas las imágenes reproducidas mediante COPY se distorsionan ligeramente alterándose en sentido vertical.

*

Las impresoras, como cualquier periférico, no deben ser conectadas al bus de cableación del Spectrum sin haber consultado previamente la documentación a ambas aparatos.

ADVANCE, retornará el equipo a la situación de impresión si se pulsa una sola vez, en caso de mantenerse presionado, se produce el avance del papel hacia arriba.

Otra posibilidad es el modo TEST, que se consigue presionando, simultáneamente, ambos botones y liberando, a continuación, el de OFF. De este manera, la impresora escribe, ininterrumpidamente, hasta que volvamos a pulsar OFF, filas de caracteres "8" y "1", con el fin de verificar el estado del papel y el contrato correcto.

Al igual que en la ZX PRINTER, un eje acoplado al motor se encarga de desplazar el papel hacia arriba por fricción, mientras se realiza la impresión. Pero hay algo que no se ha previsto: si por cualquier causa queremos rebobinar el rollo de papel, la operación será imposible, puesto que este eje no permite el giro en sentido inverso. Para realizar la impresión, utilizo una técnica denominada de líneas, es decir, se imprime una línea cada vez. Conviene aclarar que al decir línea, nos estamos refiriendo al mismo concepto que cuando hablamos de las 175 posibles en el esquema de alta resolución del archivo de imagen. De esta manera, la velocidad de impresión aumenta considerablemente, y la ALPHACOM 32 es capaz de escribir 16 líneas enteras (dos filas de caracteres completas) cada segundo, equivaliendo aproximadamente a 84 caracteres por segundo. Frente a esta rapidez considerable, el inconveniente atribuye en que empleará el mismo tiempo en imprimir, por ejemplo, una sola "A" que toda una fila de éstas. No obstante, la velocidad de impresión le sitúa como la más rápida dentro de su categoría.

El papel utilizado, nuevamente, es especial. En este caso se trata de papel térmico, es decir, recubierto por una de sus caras de sustancias termosensibles, que cambian de color cuando reciben una determinada cantidad de calor. Los resultados, en lo concerniente a calidad gráfica, y teniendo en cuenta el tamaño de la impresora, pueden considerarse excelentes. La anchura mé-

xima del papel utilizado, no debe superar los 109 mm.

La tecnología térmica supone otra de las virtudes del equipo, puesto que, a la vez de tener menos elementos mecánicos móviles, ventaja contra las averías, el nivel sonoro del proceso impresión, disminuye a límites bajísimos.

ECONOMICA DE IMPACTO

La SEIKOSHA GP-505 puede considerarse como la hermana pequeña, dentro de la amplia gama de impresoras térmicas, que esta firma japonesa ha comercializado en los últimos años. Se trata de un equipo compacto, especialmente diseñado para el ZX 81 y el ZX Spectrum. También precisa una fuente de alimentación exterior, que suministra la energía necesaria para hacer funcionar el aparato. Un interruptor ON/OFF y un indicador luminoso, sirven para verificar que la impresora se ha conectado correctamente a la alimentación. En el caso de producirse algún error, la lámpara parpadeará mientras éste no sea subsanado.

Cuando se pulsa ON, la impresora realiza una secuencia de inicialización, desplazando la cabeza de impresión dos veces a lo ancho del papel, quedando de esta manera, preparada para la escritura.

El cabezal de impresión está formado por una matriz de 7x7 pequeñas agujas, más otra adicional utilizada en gráficos de alta resolución, éstas impactan sobre la cinta entintada y el papel. La cinta va encapsulada dentro de un cassette especial, y existe la posibilidad de conseguirlo, además de en el color negro con el que se suministra el equipo, en distintos colores tales como el rojo, naranja, azul, verde, etc.

Por tratarse de una impresora térmica, el nivel de ruido aumenta considerablemente respecto a las anteriores. La velocidad de impresión es de 55 caracteres por segundo, y la calidad gráfica, aceptable. Tiene la ventaja de admitir papel normal, cuestión a tener en cuenta a la hora de pensar en el reembolso, pues los rollos de papel térmico y electrostático tienen un precio más elevado. En este caso, la anchura del papel es de 126 mm.

En resumen, todas ellas cumplen satisfactoriamente su trabajo, y tiene además en común un bajo coste, aunque lógicamente, no debemos esperar resultados excesivamente espectaculares.

La zona de memoria que el Spectrum emplea como intermediario entre la impresora y el ordenador, se denomina buffer de la impresora.



EL CUBO MAGICO



SEGURAMENTE, en alguna ocasión hemos tenido en nuestras manos el famoso CUBO DE RUBIK. Hemos dado vueltas y más vueltas a cada uno de los bloques en que está dividido, y solamente en contadas ocasiones hemos logrado reordenar satisfactoriamente los nueve cuadrados que componen cada una de las seis caras. Con este nuevo programa, tendremos plena garantía de lograr el aburrimiento.

Aunque este juego ha hecho auténtico furor entre la juventud europea, no daremos por supuesto que las nuevas generaciones conocen completamente su funcionamiento, y brindaremos unas breves explicaciones sobre el objetivo de este programa. El protagonista del juego es un cubo móvil (seis caras), compuesto por nueve cuadrados de color por cada cara, todas las caras pueden girar sobre sí mismas, en el sentido horario o antihorario.

Cada una de las caras está compuesta por nueve bloquitos de un mismo color (por supuesto diferente en cada una de ellas), que se pueden desplazar mediante un giro. El cubo inicia el juego con una zona de cada color, posteriormente se desordena de una manera aleatoria, con varios movimientos, y todo seguido, se puede pasar a ordenarlo nuevamente. Aquí llega el problema: desordenarlo es bien fácil, pero girar las caras adecuadas para que todos los colores vuelvan a quedar en su lugar, no es precisamente sencillo.

El primer movimiento se consiste en mover:



En el desarrollo del cubo que se presenta en la pantalla, la cara roja coincide con la superior.

EL CUBO DEL SPECTRUM

La representación en la pantalla de una figura con volumen es evidentemente muy compleja, de ahí que hayamos escogido como solución, representar el cubo en desarrollo, es decir, en forma de cruz, tal como se muestra en la figura. Esto implica, que la cara roja representa la parte superior, la azul clara la posterior, la amarilla la anterior, la verde la inferior, la magenta la derecha y la azul oscura la izquierda.

Una vez conocida la situación del cubo, podremos comenzar el programa. Como primera medida se nos solicita la elección del nivel de dificultad, el cual consiste en el número de movimientos aleatorios a que se someterá el cubo en su fase de desorden. Una vez escogido el nivel, el cubo se presentará en la pantalla, y se desordenará, para a continuación pasar a la fase de resolución del problema por parte del jugador.

La codificación de nuestro juego, es decir, la ma-

nera en que vamos a incidir al Spectrum el movimiento que queremos realizar, es bastante fácil de aprender, y pronto nos acostumbraremos a ella. Lo más práctico será que seguimos las indicaciones que se brindan a continuación, auxiliados por los gráficos explicativos.

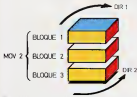
En primer lugar hemos de indicar el tipo de movimiento a efectuar. Este dependerá del plano en que deseamos realizar la jugada.

MOVIMIENTO 1 GIRO EN PLANO Z-X (anchura)
MOVIMIENTO 2 GIRO EN PLANO X-Y (altura)
MOVIMIENTO 3 GIRO EN PLANO Y-Z (profundidad)

Cada uno de estos movimientos, podría señalar el desplazamiento de cualquiera de los tres bloques que se van a ver afectados. Así pues, el siguiente paso es indicar el bloque a desplazar (de 1 a 3)

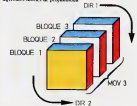
MOVIMIENTO 1 NUMERADOS DE IZQUIERDA A DERECHA
MOVIMIENTO 2 NUMERADOS DE ARRIBA A ABAJO
MOVIMIENTO 3 NUMERADOS DE PRIMER PLANO A FONDO

Finalmente, sólo queda especificar el sentido de giro del bloque escogido. Y en este caso se cum



El movimiento de rotación se efectúa en los diferentes planos.

El tercer tipo de movimiento se efectúa en los diferentes niveles de profundidad.



plie indudablemente que una imagen vale más que cien palabras, de manera que lo más sencillo será que observemos los gráficos adjuntos, donde se detalla mediante una flecha, qué sentido es el 1 y cuál el 2, según cada movimiento. Por cada jugada que ejecutamos, el ordenador efectúa una considerable cantidad de operaciones: calcula el efecto del movimiento, actualiza las seis tablas que tiene definidas, controla si el cubo ha sido solucionado, incrementa el contador de jugadas y, finalmente, muestra el movimiento especificado.

Esta serie de tareas, añadido a la estructura del programa, en forma de subrutinas, hace que el desplazamiento del cubo no sea inmediato, pero sin duda lo suficientemente ágil para lo que se pretende, sobre todo teniendo en cuenta que en este juego se confirma un conocido refrán: «Más vale malo que fuerza». Efectivamente, en el juego original siempre nos cabía la posibilidad de desmontar el cubo y volver a montarlo ordenado, sin embargo ahora.

La introducción del programa, sólo tiene el problema de las letras subrayadas que aparecen en el listado, como ya sabemos, éstas corresponden a los caracteres gráficos de las teclas afectadas. A modo orientativo, señalaremos que dichos caracteres se encuentran en las siguientes líneas: 80, 90, 130, 140, 170, 180, 6875, 6880, 6885, 6890, 6895, 6900, 6905, 6910, 6915, 6920, 6955 y 6960. Para la grabación del programa, utilizaremos el comando **SAVE "CUBO MAG,"** (sin autoejecución), o bien **SAVE "CUBO MAG," LINE 1** (con autoejecución a partir de la primera línea).



```

10 FOR A=0 TO 5:PRINT A:GOTO 10
20 FOR A=0 TO 5:PRINT A:GOTO 20
30 FOR A=0 TO 5:PRINT A:GOTO 30
40 FOR A=0 TO 5:PRINT A:GOTO 40
50 FOR A=0 TO 5:PRINT A:GOTO 50
60 FOR A=0 TO 5:PRINT A:GOTO 60
70 FOR A=0 TO 5:PRINT A:GOTO 70
80 FOR A=0 TO 5:PRINT A:GOTO 80
90 FOR A=0 TO 5:PRINT A:GOTO 90
100 FOR A=0 TO 5:PRINT A:GOTO 100
110 FOR A=0 TO 5:PRINT A:GOTO 110
120 FOR A=0 TO 5:PRINT A:GOTO 120
130 FOR A=0 TO 5:PRINT A:GOTO 130
140 FOR A=0 TO 5:PRINT A:GOTO 140
150 FOR A=0 TO 5:PRINT A:GOTO 150
160 FOR A=0 TO 5:PRINT A:GOTO 160
170 FOR A=0 TO 5:PRINT A:GOTO 170
180 FOR A=0 TO 5:PRINT A:GOTO 180
190 FOR A=0 TO 5:PRINT A:GOTO 190
200 FOR A=0 TO 5:PRINT A:GOTO 200
210 FOR A=0 TO 5:PRINT A:GOTO 210
220 FOR A=0 TO 5:PRINT A:GOTO 220
230 FOR A=0 TO 5:PRINT A:GOTO 230
240 FOR A=0 TO 5:PRINT A:GOTO 240
250 FOR A=0 TO 5:PRINT A:GOTO 250
260 FOR A=0 TO 5:PRINT A:GOTO 260
270 FOR A=0 TO 5:PRINT A:GOTO 270
280 FOR A=0 TO 5:PRINT A:GOTO 280
290 FOR A=0 TO 5:PRINT A:GOTO 290
300 FOR A=0 TO 5:PRINT A:GOTO 300
310 FOR A=0 TO 5:PRINT A:GOTO 310
320 FOR A=0 TO 5:PRINT A:GOTO 320
330 FOR A=0 TO 5:PRINT A:GOTO 330
340 FOR A=0 TO 5:PRINT A:GOTO 340
350 FOR A=0 TO 5:PRINT A:GOTO 350
360 FOR A=0 TO 5:PRINT A:GOTO 360
370 FOR A=0 TO 5:PRINT A:GOTO 370
380 FOR A=0 TO 5:PRINT A:GOTO 380
390 FOR A=0 TO 5:PRINT A:GOTO 390
400 FOR A=0 TO 5:PRINT A:GOTO 400
410 FOR A=0 TO 5:PRINT A:GOTO 410
420 FOR A=0 TO 5:PRINT A:GOTO 420
430 FOR A=0 TO 5:PRINT A:GOTO 430
440 FOR A=0 TO 5:PRINT A:GOTO 440
450 FOR A=0 TO 5:PRINT A:GOTO 450
460 FOR A=0 TO 5:PRINT A:GOTO 460
470 FOR A=0 TO 5:PRINT A:GOTO 470
480 FOR A=0 TO 5:PRINT A:GOTO 480
490 FOR A=0 TO 5:PRINT A:GOTO 490
500 FOR A=0 TO 5:PRINT A:GOTO 500
510 FOR A=0 TO 5:PRINT A:GOTO 510
520 FOR A=0 TO 5:PRINT A:GOTO 520
530 FOR A=0 TO 5:PRINT A:GOTO 530
540 FOR A=0 TO 5:PRINT A:GOTO 540
550 FOR A=0 TO 5:PRINT A:GOTO 550
560 FOR A=0 TO 5:PRINT A:GOTO 560
570 FOR A=0 TO 5:PRINT A:GOTO 570
580 FOR A=0 TO 5:PRINT A:GOTO 580
590 FOR A=0 TO 5:PRINT A:GOTO 590
600 FOR A=0 TO 5:PRINT A:GOTO 600
610 FOR A=0 TO 5:PRINT A:GOTO 610
620 FOR A=0 TO 5:PRINT A:GOTO 620
630 FOR A=0 TO 5:PRINT A:GOTO 630
640 FOR A=0 TO 5:PRINT A:GOTO 640
650 FOR A=0 TO 5:PRINT A:GOTO 650
660 FOR A=0 TO 5:PRINT A:GOTO 660
670 FOR A=0 TO 5:PRINT A:GOTO 670
680 FOR A=0 TO 5:PRINT A:GOTO 680
690 FOR A=0 TO 5:PRINT A:GOTO 690
700 FOR A=0 TO 5:PRINT A:GOTO 700
710 FOR A=0 TO 5:PRINT A:GOTO 710
720 FOR A=0 TO 5:PRINT A:GOTO 720
730 FOR A=0 TO 5:PRINT A:GOTO 730
740 FOR A=0 TO 5:PRINT A:GOTO 740
750 FOR A=0 TO 5:PRINT A:GOTO 750
760 FOR A=0 TO 5:PRINT A:GOTO 760
770 FOR A=0 TO 5:PRINT A:GOTO 770
780 FOR A=0 TO 5:PRINT A:GOTO 780
790 FOR A=0 TO 5:PRINT A:GOTO 790
800 FOR A=0 TO 5:PRINT A:GOTO 800
810 FOR A=0 TO 5:PRINT A:GOTO 810
820 FOR A=0 TO 5:PRINT A:GOTO 820
830 FOR A=0 TO 5:PRINT A:GOTO 830
840 FOR A=0 TO 5:PRINT A:GOTO 840
850 FOR A=0 TO 5:PRINT A:GOTO 850
860 FOR A=0 TO 5:PRINT A:GOTO 860
870 FOR A=0 TO 5:PRINT A:GOTO 870
880 FOR A=0 TO 5:PRINT A:GOTO 880
890 FOR A=0 TO 5:PRINT A:GOTO 890
900 FOR A=0 TO 5:PRINT A:GOTO 900
910 FOR A=0 TO 5:PRINT A:GOTO 910
920 FOR A=0 TO 5:PRINT A:GOTO 920
930 FOR A=0 TO 5:PRINT A:GOTO 930
940 FOR A=0 TO 5:PRINT A:GOTO 940
950 FOR A=0 TO 5:PRINT A:GOTO 950
960 FOR A=0 TO 5:PRINT A:GOTO 960
970 FOR A=0 TO 5:PRINT A:GOTO 970
980 FOR A=0 TO 5:PRINT A:GOTO 980
990 FOR A=0 TO 5:PRINT A:GOTO 990

```

!

Para probar el programa utilizaremos la instrucción **SAVE "CUBO MAG," LINE 1**. Si oprimimos la especificación final **LINE 1**, el programa no se autoejecutará al finalizar la línea, aunque podrá ser lanzado mediante **RUN**, sin ninguna dificultad.

*

Al introducir el programa no olvidemos apuntar los caracteres subrayados, por los gráficos definidos correspondientes a las teclas indicadas.

*

A la hora de efectuar una jugada, hemos de introducir el número correspondiente a tipo de movimiento, número de bloque y dirección en la cual el bloque va a desplazarse.

